Engineer-to-Engineer Note EE-294

ANALOG DEVICES

Technical notes on using Analog Devices DSPs, processors and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

Energy-Aware Programming on Blackfin® Processors

Contributed by Brian Mack and Seth Molloy

Rev 1 – June 26, 2007

Introduction

This EE-Note provides details regarding the Power Estimation Profiling tool introduced in the VisualDSP++® 4.5 development tools from Analog Devices.

Energy-Aware Programming is the ability to use simulation to see the relative impact of instructions, source lines of code, functions, programs, core clock frequency, and applied core voltage on the estimated energy profile of the application running on the Blackfin® processor. This allows the programmer options to make trade-offs to optimize for energy. The technique used to estimate the energy of the application is a partial implementation of a process known as *Instruction Level Energy Estimation* (ILEE).

Definitions

The following terms appear throughout this EE-Note.

Instruction Base Cost

This is the average current drawn by the core while repeatedly executing the same instruction. Its value, `I`, is obtained by direct measurement and forms the basis for the rest of the energy-estimation process.

Inter-Instruction Overhead

This is the difference between the average of two instruction base costs and the measured or estimated average current drawn for two instructions run sequentially.

Power

In the context of this document, *power* (`P`) is defined as the product of the Instruction Base Cost and the voltage applied to the core (V_{DDINT}); therefore:

$$P = I \times V_{DDINT}$$

Energy

In the context of this document, e*nergy* (`E`) is defined as the product of the power (`P`) consumed by the core and the core clock period (`T`). Since `T` is the inverse of the core clock frequency (`CCLK`):

$$E = P / CCLK$$

Instruction-Level Energy Estimation (ILEE)

Research performed with the Blackfin processor, funded by Analog Devices at Northeastern University's Computer Architecture Research Laboratory, has resulted in the integration of preliminary ILEE in the Blackfin processor simulators included in the VisualDSP++ 4.5 development tools. Currently, only instruction base cost (and not inter-instruction overhead) is being used in the estimates produced by the simulator. This reduces the accuracy of the estimates. Research is still being conducted to provide:

- The inter-instruction overhead portion of the estimate
- Refinement regarding memory configuration and activity
- Refinement regarding data-dependent factors that reduce the accuracy of the current estimates

The Northeastern University research funded by Analog Devices references *Developing Energy-Aware Strategies for the Blackfin Processor*[1].

Base cost measurements were made on an ADSP-BF533 EZ-KIT Lite® evaluation board. The database of tested instructions consists of every instruction using a subset of registers and data values.

Considerations for Special Instructions

Some instructions in the database were given special consideration because of the complexity of trying to measure power consumption during continuous execution of the same instruction. For example, sequencer return instructions like `RTI`, `RTX`, and `RTN` cannot be run in a continuous loop without creating an exception on the chip. Similarly, instructions such as `EXCPT` could not be run in the measurement environment. For these instructions, the power value of the closest instruction within the group is used. If this was not feasible, the average reading for all instructions was used.

Voltage and Frequency Scaling

As mentioned previously, power is a function of average current and voltage. Power consumption on the Blackfin processor, in turn, is a function of voltage and frequency. In order to scale power, based on the settings of voltage and frequency in the dynamic power management controller as compared to voltage and frequency settings at the time the average current is measured, the following formula is applied:

$$P_{DDDYN@V} = P_{DDDYN@V0} * (V/V0)2 * (F/F0)$$

where:

$V = V_{DDINT}$

$V0 = 1.2$ V

F = CCLK

$F0 = 250$ MHz

The above formula is discussed in *Estimating Power for ADSP-BF533 Blackfin Processors (EE-229)*[2].

Energy Units

As described in EE-229, the values in the database, if used as absolute measurements, would not be accurate enough considering factors such as leakage current, temperature, and the fabrication process of the chip, which all play a part in a the processor's power consumption. These energy estimates are to be used relative to each other, not in an absolute sense. Thus, they are referred to as instruction *Energy Units*.

Instruction Base Cost Database

Appendix A contains the instruction base cost database developed for the Blackfin processor. The following charts help to summarize the data:

- Figure 1. Average current (mA) reading per instruction group
- Figure 2. Difference (mA) between maximum and minimum current readings per instruction group

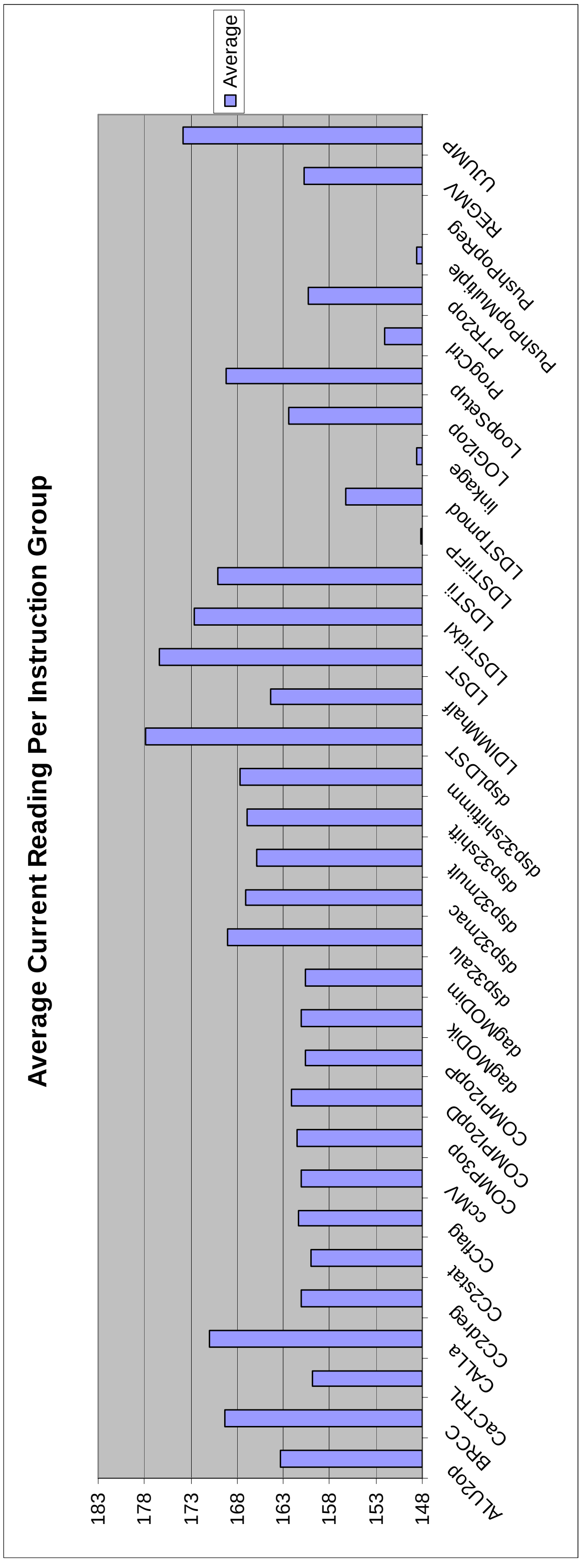

Figure 1. Average current reading (mA) per instruction group

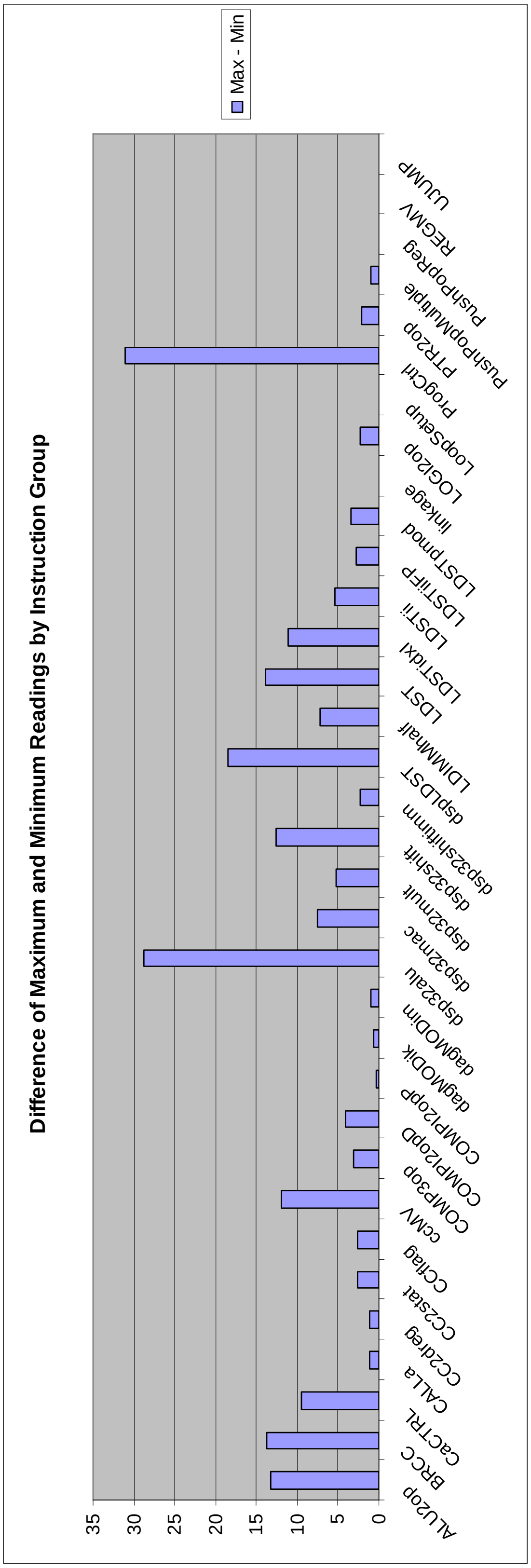

Figure 2. Difference (mA) between maximum and minimum readings by instruction group

Appendix A – Instruction Base Cost Database

Table 1 through Table 34 list instruction base cost information for unique groups of instructions.

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 = R0.B (X) ;	160.34	R0 = - R0 ;	162.301
R0 >>= R0 ;	160.745	R0 = R0.B (Z) ;	162.412
R0 = R0.L (Z) ;	160.86	R0 <<= R0 ;	162.439
R0 >>>= R0 ;	160.875	DIVS (R0 , R0) ;	162.982
R0 = (R0 + R0) << 1 ;	161.277	R0 = (R0 + R0) << 2 ;	163.296
R0 *= R0 ;	161.643	R0 = ~ R0 ;	171.121
R0 = R0.L (X) ;	162.127	DIVQ (R0 , R0) ;	173.524

Table 1. Instruction base cost for ALU2op group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
IFLUSH [P0] ;	154.478	FLUSHINV [P0] ;	161.492
IFLUSH [P0 ++] ;	155.149	PREFETCH [P0 ++] ;	161.629
FLUSH [P0] ;	159.669	FLUSHINV [P0 ++] ;	163.097
PREFETCH [P0] ;	159.941	FLUSH [P0 ++] ;	163.926

Table 2. Instruction base cost for CaCTRL group

Instruction	Instruction Base Cost (mA)
IF CC JUMP 0 /*0x0*/ ;	161.742
IF !CC JUMP 0 /*0x0*/ ;	168.835
IF CC JUMP 0 /*0x0*/ (BP) ;	171.236
IF !CC JUMP 0 /*0x0*/ (BP) ;	175.513

Table 3. Instruction base cost for BRCC group

Instruction	Instruction Base Cost (mA)
CALL 0 /*0x0*/ ;	170.378
JUMP.L 0 /*0x0*/ ;	171.584

Table 4. Instruction base cost for CALLa group

Instruction	Instruction Base Cost (mA)
R0 = CC ;	160.717
CC = !CC ;	160.857
CC = R0 ;	161.801

Table 5. Instruction base cost for CC2dreg group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
CC |= AZ ;	158.552	AZ = CC ;	160.587
CC &= AZ ;	158.686	AZ ^= CC ;	160.688
CC = AZ ;	158.832	AZ &= CC ;	160.73
AZ |= CC ;	160.57	CC ^= AZ ;	161.144

Table 6. Instruction base cost for CC2stat group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
CC = P0 <= 0x0 (IU) ;	160.045	CC = P0 <= 0 ;	161.461
CC = P0 < 0 ;	160.162	CC = R0 <= R0 (IU) ;	161.629
CC = P0 < 0x0 (IU) ;	160.24	CC = R0 < 0 ;	161.653
CC = P0 < P0 ;	160.305	CC = A0 < A1 ;	161.85
CC = A0 == A1 ;	160.428	CC = R0 < R0 (IU) ;	162.036
CC = R0 == R0 ;	160.587	CC = R0 < R0 ;	162.071
CC = A0 <= A1 ;	160.595	CC = R0 < 0x0 (IU) ;	162.124
CC = P0 < P0 (IU) ;	160.746	CC = R0 == 0 ;	162.544
CC = P0 <= P0 ;	160.965	CC = R0 <= 0x0 (IU) ;	162.612
CC = P0 == P0 ;	161.175	CC = R0 <= R0 ;	162.651
CC = P0 == 0 ;	161.347	CC = R0 <= 0 ;	162.671
CC = P0 <= P0 (IU) ;	161.43		

Table 7. Instruction base cost for CCflag group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
IF !CC P0 = P0 ;	153.377	IF CC P0 = R0 ;	163.935
IF CC P0 = P0 ;	153.414	IF !CC P0 = R0 ;	164.07
IF !CC R0 = R0 ;	160.621	IF CC R0 = P0 ;	165.058
IF CC R0 = R0 ;	162.616	IF !CC R0 = P0 ;	165.358

Table 8. Instruction base cost for ccMV group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
P0 = P0 + (P0 << 2) ;	159.749	P0 = P0 << 1 ;	161.399
P0 = P0 + (P0 << 1) ;	161.018	R0 = R0 ^ R0 ;	162.231
R0 = R0 + R0 ;	161.102	R0 = R0 | R0 ;	162.636
R0 = R0 - R0 ;	161.164	R0 = R0 & R0 ;	162.807

Table 9. Instruction base cost for COMP3op group

Instruction	Instruction Base Cost (mA)
R0 = 0 ;	160.072
R0 += 0 ;	164.194

Table 10. Instruction base cost for COMPI2opD group

Instruction	Instruction Base Cost (mA)
P0 = 0 ;	160.5
P0 += 0 ;	160.766

Table 11. Instruction base cost for COMPI2opP group

Instruction	Instruction Base Cost (mA)
I0 -= 4 ;	160.819
I0 += 4 ;	161.118
I0 += 2 ;	161.155
I0 -= 2 ;	161.456

Table 12. Instruction base cost for dagMODik group

Instruction	Instruction Base Cost (mA)
I0 += M0 (BREV) ;	160.006
I0 += M0 ;	160.9
I0 -= M0 ;	160.997

Table 13. Instruction base cost for dagMODim group

Instruction	Base Cost (mA)	Instruction	Base Cost (mA)
(R1, R0) = SEARCH R0 (GE);	157.263	R0 = MAX (R0 , R0) ;	168.844
(R1, R0) = SEARCH R0 (GT);	157.472	R0.H = R0.L + R0.L (S) ;	168.878
(R1, R0) = SEARCH R0 (LE);	158.326	R0 = BYTEOP2P (R1:0 , R1:0) (TH , R) ;	168.932
(R1, R0) = SEARCH R0 (LT);	158.327	R0.H = R0.L + R0.L (NS) ;	168.95
DISALGNEXCPT;	164.87	R0 = BYTEOP3P (R1:0 , R1:0) (LO) ;	168.955
A1 = A0 = 0;	165.074	R0 = BYTEOP2P (R1:0 , R1:0) (RNDH , R) ;	168.979
A0 = 0;	165.405	R0 = MIN (R0 , R0) (V) ;	168.997
R0.L = A1.x;	165.452	R0 = MIN (R0 , R0) ;	169.045
A1 = 0;	165.492	R0 = BYTEOP2P (R1:0 , R1:0) (RNDL) ;	169.104
(R1, R0) = BYTEUNPACK R1:0;	165.539	R0.H = (A0 += A1) ;	169.129
R0.L = A0.x ;	165.674	R0.L = R0.H + R0.L (S) ;	169.17
A1.h = R0.H ;	165.853	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (CO) ;	169.251
A1.x = R0.L;	165.892	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASL , CO) ;	169.264
A0.x = R0.L;	165.92	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASL , SCO) ;	169.269
R0 = BYTEPACK (R0 , R0);	166.051	R0.H = R0.H + R0.H (S) ;	169.329
A1.l = R0.L;	166.086	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASR , SCO) ;	169.337
(R1 , R0) = BYTEUNPACK R1:0 (R);	166.109	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (S) ;	169.382
A1 = ABS A0;	166.184	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASL) ;	169.384
A0.L = R0.L;	166.276	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASR , S) ;	169.396
R0.H=R0.L=SIGN(R0.H)*R0.H+SIGN(R0.L)*R0.L;	166.334	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASL , SCO) ;	169.416
A0.h = R0.H;	166.456	R0 = BYTEOP2P (R1:0 , R1:0) (TH) ;	169.418
A0 = ABS A1;	166.587	R0 = R0 +|+ R0 (CO) ;	169.421
A1 = - A0;	166.682	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASL) ;	169.431
A1 = A1 (S);	166.689	R0 = BYTEOP3P (R1:0 , R1:0) (HI) ;	169.451
A1 = ABS A1;	166.712	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASL , CO) ;	169.46
A0 = - A1;	166.724	R0 = R0 +|+ R0 ;	169.48
A0 = A0 (S);	166.725	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASR , SCO);	169.526
(R1 , R0) = BYTEOP16P (R1:0 , R1:0);	166.89	R0 = BYTEOP2P (R1:0 , R1:0) (TL) ;	169.546
(R1 , R0) = BYTEOP16M (R1:0 , R1:0);	166.94	R1 = R0 + R0 , R0 = R0 - R0 (S) ;	169.564
R0.H = R0 (RND);	167.006	R0.L = R0.L + R0.H (S) ;	169.573
(R1 , R0) = BYTEOP16M (R1:0 , R1:0) (R);	167.044	R1 = R0 + R0 , R0 = R0 - R0 ;	169.582
R0.L = R0 (RND);	167.051	R0 = BYTEOP2P (R1:0 , R1:0) (RNDH) ;	169.582
(R1 , R0) = BYTEOP16P (R1:0 , R1:0) (R) ;	167.058	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASR , CO);	169.585
R0.L = R0 - R0 (RND20) ;	167.072	R1 = R0 +|- R0 , R0 = R0 -|+ R0 ;	169.631
R0.L = R0.H - R0.H (S) ;	167.147	R0.L = R0.L + R0.L (S) ;	169.657
R0 = BYTEOP1P (R1:0 , R1:0) (R) ;	167.211	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASR , CO);	169.668
R0.L = R0.H - R0.H (NS) ;	167.244	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASL , S) ;	169.676

R0 = R0 - R0 (NS) ;	167.254	R0.L = (A0 += A1) ;	169.803
R0.H = R0.H - R0.H (NS) ;	167.258	R0.H = R0.H + R0.L (S) ;	169.814
R0.L = R0.L + R0.L (NS) ;	167.259	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (SCO) ;	169.886
R0.H = R0.H + R0.H (NS) ;	167.321	R0.H = R0 + R0 (RND12) ;	169.894
R0.L = R0.L - R0.L (NS) ;	167.414	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (CO) ;	170.017
R0 = BYTEOP1P (R1:0 , R1:0) (T , R) ;	167.445	R0.L = R0 + R0 (RND12) ;	170.019
R0 = R0 + R0 (NS) ;	167.529	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (ASR) ;	170.02
R0 = R0 - R0 (S) ;	167.597	R0.H = R0.L + R0.H (S) ;	170.104
R0.H = R0.H - R0.H (S) ;	167.645	R0 = BYTEOP3P (R1:0 , R1:0) (HI , R) ;	170.186
R0 = ABS R0 (V) ;	167.646	R1 = R0 +|+ R0 , R0 = R0 -|- R0 (SCO) ;	170.191
A0 = ABS A0 ;	167.656	R1 = A0 + A1 , R0 = A0 - A1 ;	170.213
R0.H = R0.L - R0.L (S) ;	167.668	R1 = A1 + A0 , R0 = A1 - A0 (S) ;	170.262
R0.H = R0 - R0 (RND20) ;	167.796	R1 = A1 + A0 , R0 = A1 - A0 ;	170.341
R0.L = R0.L - R0.L (S) ;	167.82	R1 = A0 + A1 , R0 = A0 - A1 (S) ;	170.353
R0.L = R0.H + R0.H (S) ;	167.857	R0.L = R0.H - R0.L (S) ;	170.488
R0.L = R0 - R0 (RND12) ;	167.857	R0 = BYTEOP3P (R1:0 , R1:0) (LO , R) ;	170.544
R0.L = R0.H + R0.H (NS) ;	167.859	R0 = (A0 += A1) ;	170.613
R0 = BYTEOP1P (R1:0 , R1:0) ;	167.885	R1 = R0 +|+ R0 , R0 = R0 -|- R0 ;	170.657
R0.H = R0.L - R0.L (NS) ;	167.906	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASR , S) ;	170.712
R0 = BYTEOP1P (R1:0 , R1:0) (T) ;	167.929	R0 = R0 +|- R0 (S) ;	170.758
A0 -= A1 (W32) ;	167.954	R0.L = R0.H - R0.L (NS) ;	170.774
R0.H = R0 + R0 (RND20) ;	167.961	R0 = R0 -|+ R0 (S) ;	170.777
A0 += A1 (W32) ;	167.972	R0 = R0 + R0 (S) ;	170.828
A0 -= A1 ;	167.993	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASR) ;	171.094
R0 = R0 -|- R0 ;	168.01	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (ASL , S) ;	171.791
A0 += A1 ;	168.11	R1 = R0 +|- R0 , R0 = R0 -|+ R0 (S) ;	172.376
R0.L = R0 + R0 (RND20) ;	168.151	R0.L = R0.L + R0.H (NS) ;	172.956
R0 = ABS R0 ;	168.163	R0 = R0 +|+ R0 (SCO) ;	173.121
A1 = ABS A1 , A0 = ABS A0 ;	168.17	R0 = R0 +|+ R0 (S) ;	173.299
R0 = R0 -|+ R0 (SCO) ;	168.2	R0.L = R0.H + R0.L (NS) ;	173.537
A1 = A1 (S) , A0 = A0 (S) ;	168.23	R0.L = R0.L - R0.H (NS) ;	173.61
R0 = R0 -|+ R0 ;	168.284	R0.H = R0.H + R0.L (NS) ;	173.624
R0.H = R0 - R0 (RND12) ;	168.295	R0.H = R0.H - R0.L (NS) ;	173.7
R0 = R0 -|- R0 (CO) ;	168.303	SAA (R1:0 , R1:0) ;	173.895
R0 = R0 -|- R0 (S) ;	168.324	R0.H = R0.L - R0.H (NS) ;	174
R0.H = R0.H - R0.L (S) ;	168.344	SAA (R1:0 , R1:0) (R) ;	174.208
R0 = R0 +|- R0 (SCO) ;	168.415	R0.H = R0.L - R0.H (S) ;	174.352
R0 = R0 -|- R0 (SCO) ;	168.417	R0.H = R0.L + R0.H (NS) ;	174.716
R0 = BYTEOP2P (R1:0 , R1:0) (TL , R) ;	168.494	A1 = - A1 ;	175.622
R0 = MAX (R0 , R0) (V) ;	168.578	A0 = - A0 ;	176.495
R0 = R0 +|- R0 ;	168.587	A1 = - A1 , A0 = - A0 ;	180.12

R0.L = R0.L - R0.H (S) ;	168.618	R0 = - R0 (NS) ;	181.982
R0 = R0 +|- R0 (CO) ;	168.695	R0 = - R0 (S) ;	182.121
R0 = BYTEOP2P (R1:0 , R1:0) (RNDL , R) ;	168.7	R0 = - R0 (V) ;	186.032
R0 = R0 -|+ R0 (CO) ;	168.749		

Table 14. Instruction base cost for dsp32alu group

Instruction	Base Cost (mA)	Instruction	Base Cost (mA)
R0.L = (A0 = R0.L * R0.L) (IS) ;	163.266	R0.H = A1 (S2RND) ;	167.653
R0.L = (A0 = R0.L * R0.L) (IU) ;	163.298	R0.L = A0 ;	167.657
R0.L = (A0 -= R0.L * R0.L) (TFU) ;	163.301	R0.L = A0 (FU) ;	167.688
R0.L = (A0 = R0.L * R0.L) (S2RND) ;	163.304	R0.H = (A1 = R0.L * R0.L) (IS) ;	167.705
R0.L = (A0 += R0.L * R0.L) (S2RND) ;	163.348	A0 += R0.L * R0.L (W32) ;	167.707
R0.L = (A0 += R0.L * R0.L) (T) ;	163.384	A1 += R0.L * R0.L (M , W32) ;	167.727
R0.L = (A0 -= R0.L * R0.L) ;	163.401	R0.H = (A1 = R0.L * R0.L) (S2RND) ;	167.73
R0.L = (A0 = R0.L * R0.L) (TFU) ;	163.417	A1 += R0.L * R0.L (W32) ;	167.737
R0.L = (A0 = R0.L * R0.L) (IH) ;	163.431	R0.H = (A1 = R0.L * R0.L) (M , S2RND);	167.748
R0.L = (A0 += R0.L * R0.L) (TFU) ;	163.44	A1 += R0.L * R0.L (IS) ;	167.755
R0.L = (A0 -= R0.L * R0.L) (T) ;	163.508	A1 = R0.L * R0.L ;	167.76
R0.L = (A0 -= R0.L * R0.L) (FU) ;	163.527	A1 -= R0.L * R0.L (IS) ;	167.772
R0.L = (A0 -= R0.L * R0.L) (IU) ;	163.64	A1 -= R0.L * R0.L ;	167.775
R0.L = (A0 -= R0.L * R0.L) (S2RND) ;	163.668	A1 += R0.L * R0.L (M) ;	167.789
R0.L = (A0 += R0.L * R0.L) (IH) ;	163.716	R0.H = (A1 = R0.L * R0.L) (M , IH) ;	167.79
R0.L = (A0 = R0.L * R0.L) (FU) ;	163.722	R0.L = A0 (IU) ;	167.79
R0.L = (A0 += R0.L * R0.L) (FU) ;	163.733	R0.H = (A1 = R0.L * R0.L) (ISS2) ;	167.791
R0.L = (A0 = R0.L * R0.L) (ISS2) ;	163.75	R0.H = A1 ;	167.799
R0.L = (A0 = R0.L * R0.L) (T) ;	163.801	A1 -= R0.L * R0.L (FU) ;	167.799
R0.L = (A0 -= R0.L * R0.L) (IH) ;	163.821	A1 += R0.L * R0.L ;	167.819
R0.L = (A0 += R0.L * R0.L) ;	163.891	R0.H = A1 (IH) ;	167.827
R0.L = (A0 -= R0.L * R0.L) (IS) ;	163.959	A0 -= R0.L * R0.L (FU) ;	167.836
R0.L = (A0 = R0.L * R0.L) ;	164.355	A0 += R0.L * R0.L (IS) ;	167.853
R0.L = (A0 -= R0.L * R0.L) (ISS2) ;	164.524	R0.L = A0 (S2RND) ;	167.856
R0 = (A0 = R0.L * R0.L) , A1 -= R0.L * R0.L (M);	164.553	R0.H = (A1 = R0.L * R0.L) (FU) ;	167.857
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (IS);	164.629	A0 = R0.L * R0.L (W32) ;	167.91
R0 = (A0 = R0.L * R0.L), A1 -= R0.L * R0.L (ISS2);	164.709	A0 += R0.L * R0.L (FU) ;	167.931
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (M);	164.781	R0.L = A0 (IH) ;	167.932
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (FU);	164.804	A0 += R0.L * R0.L ;	167.944
R0.H=(A1=R0.L*R0.L)(M), R0.L=(A0=R0.L*R0.L);	164.808	R0.H = A1 (T) ;	167.972
R0 = (A0 = R0.L * R0.L), A1 -= R0.L * R0.L (M, IS);	164.821	R0.L = A0 (ISS2) ;	167.976
R0 = (A0 = R0.L * R0.L), A1 -= R0.L * R0.L (FU) ;	164.876	R0.H = (A1 = R0.L * R0.L) (IH) ;	167.982
R0 = (A0=R0.L * R0.L), A1 += R0.L * R0.L (M, FU);	164.884	A0 -= R0.L * R0.L (W32) ;	168.03
R0 = (A0 = R0.L * R0.L), A1 = R0.L * R0.L (FU);	164.905	R0.H = A1 (IS) ;	168.03
R0 = (A0 = R0.L * R0.L), A1 = R0.L * R0.L(M , FU);	164.921	R0.L = A0 (IS) ;	168.124
R0=(A0=R0.L*R0.L), A1 -= R0.L*R0.L(M, S2RND);	164.946	R0.H = (A1 = R0.L * R0.L) ;	168.178

R0 = (A0 = R0.L * R0.L) , A1 -= R0.L * R0.L (IS);	164.954	R0.H = A1 (IU) ;	168.324
R0.H=(A1=R0.L*R0.L), R0.L=(A0=R0.L*R0.L) (IS);	164.984	R0.H = A1 (FU) ;	168.351
R0.L = (A0 = R0.L * R0.L), A1 = R0.L * R0.L (FU);	164.998	R0.H = (A1 -= R0.L * R0.L) (ISS2) ;	168.353
R0 = (A0=R0.L*R0.L), A1 = R0.L * R0.L (S2RND);	164.999	R0.H = (A1 += R0.L * R0.L) (M , ISS2) ;	168.373
R0 = (A0 = R0.L * R0.L) , A1 += R0.L * R0.L ;	165.009	R0.H = (A1 += R0.L * R0.L) (M , IU) ;	168.406
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (IS) ;	165.017	R0.H = (A1 -= R0.L * R0.L) (M , IS) ;	168.434
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L ;	165.046	R0.H = (A1 -= R0.L * R0.L) (M , IU) ;	168.442
R0 = (A0=R0.L*R0.L), A1 -= R0.L*R0.L (M, ISS2);	165.056	R0.H = (A1 += R0.L * R0.L) (ISS2) ;	168.45
R0 = (A0=R0.L*R0.L), A1 -= R0.L * R0.L (S2RND);	165.065	R0.H = A1 (ISS2) ;	168.511
R0 = (A0=R0.L*R0.L) , A1 += R0.L * R0.L (FU) ;	165.068	A0 = R0.L * R0.L ;	168.526
R0 = (A0=R0.L*R0.L), A1=R0.L*R0.L (M, S2RND);	165.104	R0.H = (A1 -= R0.L * R0.L) (M, S2RND);	168.527
R0 = (A0=R0.L*R0.L), A1 -= R0.L * R0.L (M, FU);	165.117	R0.H = (A1 -= R0.L * R0.L) (IS) ;	168.591
R0 = (A0=R0.L*R0.L), A1 += R0.L * R0.L(M, ISS2);	165.123	R0.H = (A1 += R0.L * R0.L) (IS) ;	168.629
R0.H=(A1=R0.L*R0.L)(M),R0.L=(A0=R0.L*R0.L)(TFU);	165.134	A0 = R0.L * R0.L (IS) ;	168.665
R0 = (A0=R0.L*R0.L), A1 -= R0.L * R0.L ;	165.145	R0.H = (A1 += R0.L * R0.L) (M , IS) ;	168.697
R0.H = (A1=R0.L*R0.L)(M), R0.L = (A0=R0.L*R0.L)(IH);	165.165	R0.H = (A1 += R0.L * R0.L) (M , FU) ;	168.71
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L);	165.177	R0.H = (A1 += R0.L * R0.L) (M , S2RND);	168.711
R0 = (A0=R0.L*R0.L), A1 += R0.L * R0.L (S2RND);	165.206	R0.H = (A1 -= R0.L * R0.L) ;	168.716
R0 = (A0=R0.L*R0.L), A1 = R0.L * R0.L (M , ISS2);	165.22	R0.H = (A1 += R0.L * R0.L) (M , TFU) ;	168.717
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (M , IS);	165.23	R0.H = (A1 += R0.L * R0.L) (M) ;	168.736
R0 = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (ISS2);	165.247	R0.H = (A1 -= R0.L * R0.L) (M , TFU) ;	168.784
R0 = (A0 = R0.L * R0.L) , A1 += R0.L * R0.L (M , IS);	165.256	R0.H = (A1 -= R0.L * R0.L) (S2RND) ;	168.815
R0 = (A0 = R0.L * R0.L) , A1 += R0.L * R0.L (M);	165.267	A0 -= R0.L * R0.L ;	168.821
R0 = (A0 = R0.L * R0.L) , A1 += R0.L * R0.L (ISS2);	165.268	R0.H = (A1 -= R0.L * R0.L) (M , ISS2) ;	168.83
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L)(TFU);	165.271	R0.H = (A1 += R0.L * R0.L) (T) ;	168.856
R0.L = (A0 = R0.L * R0.L), A1 -= R0.L * R0.L (S2RND);	165.273	R0.H = (A1 -= R0.L * R0.L) (M , FU) ;	168.896
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L)(T);	165.304	R0.H = (A1 -= R0.L * R0.L) (M , T) ;	168.905
MNOP;	165.315	R0.H = (A1 += R0.L * R0.L) ;	168.926
R0.H =(A1=R0.L*R0.L)(M), R0.L= (A0=R0.L*R0.L)(FU);	165.325	R0.H = (A1 -= R0.L * R0.L) (M , IH) ;	168.935
R0 = (A0 = R0.L * R0.L), A1 = R0.L * R0.L;	165.344	A0 -= R0.L * R0.L (IS) ;	168.963
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L) (IU);	165.353	R0.H = (A1 += R0.L * R0.L) (M , T) ;	168.982
R0.H = (A1=R0.L*R0.L)(M), R0.L = (A0=R0.L*R0.L)(IU);	165.354	R0.H = (A1 -= R0.L * R0.L) (M) ;	168.988
R1 = (A1 = R0.L * R0.L), R0 = (A0 = R0.L * R0.L) (FU);	165.407	R0.H = (A1 += R0.L * R0.L) (S2RND) ;	168.993
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L) (FU);	165.429	A1=R0.L*R0.L (M), A0=R0.L*R0.L (W32);	169.003
R0.H = (A1=R0.L*R0.L)(M), R0.L = (A0=R0.L*R0.L)(T);	165.438	R0.H = (A1 -= R0.L * R0.L) (T) ;	169.032
R0 = (A0 = R0.L * R0.L) , A1 += R0.L * R0.L (IS);	165.445	R0.H = (A1 -= R0.L * R0.L) (IH) ;	169.096
R0 = (A0=R0.L*R0.L), A1 += R0.L * R0.L (M, S2RND);	165.489	R0.H = (A1 += R0.L * R0.L) (M , IH) ;	169.156
R0.H = (A1=R0.L*R0.L), R0.L = (A0=R0.L*R0.L) (IH);	165.505	A1= R0.L * R0.L, A0= R0.L * R0.L (W32);	169.21
R0.H=(A1=R0.L*R0.L), R0.L=(A0=R0.L*R0.L)(S2RND);	165.506	A0=R0.L*R0.L, R1=(A1=R0.L*R0.L) (FU);	169.236
R0.H=(A1=R0.L*R0.L)(M),R0.L=(A0=R0.L*R0.L)(ISS2);	165.532	R0.H = (A1 += R0.L * R0.L) (IH) ;	169.344
R1 = (A1=R0.L*R0.L), R0 = (A0=R0.L*R0.L) (S2RND);	165.585	A1 = R0.L * R0.L , A0 = R0.L * R0.L (FU);	169.404

R1 = (A1=R0.L*R0.L) (M), R0 = (A0=R0.L*R0.L) (IS);	165.589	A0=R0.L*R0.L,R0.H=(A1=R0.L*R0.L)(FU);	169.577
R0.H=(A1=R0.L*R0.L) , R0.L=(A0=R0.L*R0.L) (ISS2);	165.592	A0 = R0.L * R0.L , R1 = (A1 = R0.L * R0.L);	169.773
R0.L = (A0 += R0.L * R0.L) (IU);	165.613	A0=R0.L * R0.L , R1 = (A1=R0.L*R0.L)(IS);	169.826
MNOP ;	165.619	R0.H = (A1 -= R0.L * R0.L) (FU) ;	169.921
R0.L = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L (IS);	165.652	R0.H = (A1 += R0.L * R0.L) (TFU) ;	169.931
R0.L = (A0 += R0.L * R0.L) (ISS2);	165.716	R0.H = A1 , R0.L = A0 ;	170.077
R1=(A1=R0.L*R0.L)(M), R0=(A0=R0.L*R0.L)(S2RND);	165.719	A1 = R0.L * R0.L , A0 = R0.L * R0.L ;	170.081
R0.H = (A1=R0.L*R0.L)(M), R0.L = (A0=R0.L*R0.L)(IS);	165.729	R0.H = (A1 -= R0.L * R0.L) (IU) ;	170.094
R0.L = (A0 += R0.L * R0.L) (IS);	165.734	R0.H = (A1 -= R0.L * R0.L) (TFU) ;	170.103
R1 = (A1=R0.L*R0.L) (M), R0 = (A0=R0.L*R0.L) (FU);	165.752	R0.H = A1 , R0.L = A0 (IH) ;	170.133
R1 = (A1=R0.L*R0.L) (M), R0 = (A0=R0.L*R0.L) (ISS2);	165.755	R0.H = (A1 += R0.L * R0.L) (FU) ;	170.151
R0.H=(A1=R0.L*R0.L)(M),R0.L=(A0=R0.L*R0.L)(S2RND);	165.814	A0=R0.L*R0.L,R1=(A1+=R0.L*R0.L)(ISS2);	170.151
R1 = (A1 = R0.L * R0.L) (M), R0 = (A0 = R0.L * R0.L);	165.842	A0 = R0.L * R0.L , R1 = (A1+=R0.L * R0.L);	170.179
R1 = (A1 = R0.L * R0.L) , R0 = (A0 = R0.L * R0.L) (IS);	165.869	A1 = R0.L * R0.L (M) , A0 = R0.L * R0.L ;	170.223
R1 = (A1 = R0.L * R0.L) , R0 = (A0 = R0.L * R0.L);	165.87	A0=R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(FU);	170.233
R0.L = (A0 = R0.L * R0.L) , A1 = R0.L * R0.L;	165.937	A0=R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(IS);	170.239
R1 = (A1 = R0.L * R0.L), R0 = (A0 = R0.L * R0.L) (ISS2);	165.937	R0.H = A1 , R0.L = A0 (T) ;	170.248
A1 = R0.L * R0.L (FU);	167.148	R0.H = (A1 += R0.L * R0.L) (IU) ;	170.283
R0.H = (A1 = R0.L * R0.L) (M , TFU);	167.287	A0 += R0.L * R0.L , R1 = (A1+=R0.L*R0.L);	170.287
A1 += R0.L * R0.L (FU);	167.3	A0=R0.L*R0.L, R1=(A1+=R0.L*R0.L)(S2RND);	170.302
A1 = R0.L * R0.L (M , W32);	167.355	A0+=R0.L*R0.L, R1=(A1+=R0.L*R0.L)(IS);	170.311
A1 = R0.L * R0.L (IS) ;	167.369	A0 -= R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(S2RND);	170.345
A1 = R0.L * R0.L (W32) ;	167.372	A1 = R0.L * R0.L , A0 = R0.L * R0.L (IS) ;	170.362
A1 = R0.L * R0.L (M) ;	167.466	R0.H = A1 , R0.L = A0 (S2RND) ;	170.38
R0.H = (A1 = R0.L * R0.L) (M , T) ;	167.468	A0 = R0.L * R0.L , R0.H = (A1=R0.L*R0.L);	170.446
A1 -= R0.L * R0.L (W32) ;	167.502	R0.H = A1 , R0.L = A0 (FU) ;	170.525
R0.H = (A1 = R0.L * R0.L) (IU) ;	167.517	A0 -= R0.L * R0.L, R1 = (A1+=R0.L*R0.L);	170.566
R0.H = (A1 = R0.L * R0.L) (M , IU) ;	167.528	A0 -= R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(ISS2);	170.57
R0.H = (A1 = R0.L * R0.L) (M , FU) ;	167.551	A0+=R0.L*R0.L, R1=(A1+=R0.L*R0.L)(FU);	170.612
R0.H = (A1 = R0.L * R0.L) (M , ISS2) ;	167.557	A0 += R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(ISS2);	170.629
A1 -= R0.L * R0.L (M , W32) ;	167.575	A0-=R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(IS);	170.63
A0 = R0.L * R0.L (FU) ;	167.582	R0.H = A1 , R0.L = A0 (IU) ;	170.632
R0.H = (A1 = R0.L * R0.L) (TFU) ;	167.596	A0-=R0.L*R0.L, R1=(A1+=R0.L*R0.L)(FU);	170.634
R0.H = (A1 = R0.L * R0.L) (T) ;	167.603	A0 += R0.L*R0.L, R1 = (A1+=R0.L*R0.L)(S2RND);	170.639
R0.H = (A1 = R0.L * R0.L) (M , IS) ;	167.615	A0=R0.L*R0.L, R0.H = (A1=R0.L*R0.L)(IS);	170.672
R0.H = (A1 = R0.L * R0.L) (M) ;	167.624	R0.H = A1 , R0.L = A0 (ISS2) ;	170.763
R0.L = A0 (T) ;	167.648	R0.H = A1 , R0.L = A0 (IS) ;	170.838

Table 15. Instruction base cost for dsp32mac group

Instruction	Base Cost (mA)	Instruction	Base Cost (mA)
R0.L = R0.L * R0.L (TFU) ;	162.81	R1 = R0.L * R0.L (M) , R0 = R0.L * R0.L (IS) ;	165.327
R0.L = R0.L * R0.L (IH) ;	163.021	R1 = R0.L * R0.L , R0 = R0.L * R0.L (FU) ;	165.344
R0.L = R0.L * R0.L (ISS2) ;	163.05	R1 = R0.L * R0.L , R0 = R0.L * R0.L (IS) ;	165.4
R0.L = R0.L * R0.L (S2RND) ;	163.066	R1 = R0.L * R0.L (M) , R0 = R0.L * R0.L ;	165.407
R0.L = R0.L * R0.L (FU) ;	163.126	R1 = R0.L * R0.L , R0 = R0.L * R0.L (ISS2) ;	165.411
R0 = R0.L * R0.L ;	163.31	R1 = R0.L * R0.L , R0 = R0.L * R0.L (S2RND) ;	165.447
R0.L = R0.L * R0.L (IU) ;	163.386	R0.H = R0.L * R0.L (M) ;	167.321
R0 = R0.L * R0.L (S2RND) ;	163.396	R1 = R0.L * R0.L (FU) ;	167.355
R0 = R0.L * R0.L (FU) ;	163.433	R1 = R0.L * R0.L (M , IS) ;	167.371
R0 = R0.L * R0.L (ISS2) ;	163.624	R1 = R0.L * R0.L (S2RND) ;	167.482
R0.L = R0.L * R0.L (T) ;	163.701	R0.H = R0.L * R0.L (IS) ;	167.499
R0 = R0.L * R0.L (IS) ;	163.775	R0.H = R0.L * R0.L (M , TFU) ;	167.52
R0.L = R0.L * R0.L ;	164.308	R0.H = R0.L * R0.L (T) ;	167.525
R0.L = R0.L * R0.L (IS) ;	164.353	R0.H = R0.L * R0.L (M , T) ;	167.544
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (TFU) ;	164.44	R0.H = R0.L * R0.L ;	167.553
R0.H = R0.L * R0.L (M), R0.L = R0.L * R0.L (IU);	164.517	R1 = R0.L * R0.L (M) ;	167.555
R0.H = R0.L * R0.L (M), R0.L = R0.L * R0.L (TFU);	164.579	R0.H = R0.L * R0.L (M , ISS2) ;	167.57
R0.H = R0.L * R0.L (M) , R0.L = R0.L * R0.L (IS) ;	164.611	R0.H = R0.L * R0.L (M , FU) ;	167.583
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (S2RND);	164.638	R0.H = R0.L * R0.L (M , IU) ;	167.614
R0.H = R0.L * R0.L (M), R0.L = R0.L * R0.L (FU);	164.676	R0.H = R0.L * R0.L (TFU) ;	167.619
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (T) ;	164.76	R0.H = R0.L * R0.L (FU) ;	167.623
R0.H=R0.L*R0.L(M), R0.L = R0.L * R0.L(S2RND);	164.779	R0.H = R0.L * R0.L (M , IH) ;	167.714
R0.H = R0.L * R0.L (M), R0.L = R0.L * R0.L (IH);	164.81	R1 = R0.L * R0.L (IS) ;	167.738
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (IU) ;	164.811	R1 = R0.L * R0.L (M , FU) ;	167.766
R0.H = R0.L * R0.L(M), R0.L = R0.L * R0.L(ISS2);	164.822	R0.H = R0.L * R0.L (IU) ;	167.823
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (ISS2) ;	164.95	R0.H = R0.L * R0.L (M , S2RND) ;	167.83
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (FU) ;	164.988	R0.H = R0.L * R0.L (ISS2) ;	167.842
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (IH) ;	164.989	R0.H = R0.L * R0.L (IH) ;	167.856
R0.H = R0.L * R0.L (M) , R0.L = R0.L * R0.L ;	165.083	R0.H = R0.L * R0.L (M , IS) ;	167.871
R1 = R0.L * R0.L , R0 = R0.L * R0.L ;	165.112	R1 = R0.L * R0.L (M , ISS2) ;	167.914
R1 = R0.L * R0.L (M) , R0 = R0.L * R0.L (ISS2) ;	165.118	R1 = R0.L * R0.L (ISS2) ;	167.964
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L (IS) ;	165.187	R0.H = R0.L * R0.L (S2RND) ;	167.989
R0.H = R0.L * R0.L (M), R0.L = R0.L * R0.L (T);	165.23	R1 = R0.L * R0.L (M , S2RND) ;	168
R0.H = R0.L * R0.L , R0.L = R0.L * R0.L ;	165.235	R1 = R0.L * R0.L (M , S2RND) ;	168.002
R1 = R0.L * R0.L (M) , R0 = R0.L * R0.L (FU) ;	165.282	R1 = R0.L * R0.L ;	168.103

Table 16. Instruction base cost for dsp32mult group

Instruction	Base Cost (mA)	Instruction	Base Cost (mA)
R0 = ALIGN16 (R0 , R0) ;	162.525	R0.H = ASHIFT R0.L BY R0.L ;	167.02
R0 = ALIGN24 (R0 , R0) ;	162.784	R0.L = ASHIFT R0.L BY R0.L (S) ;	167.027
R0 = ALIGN8 (R0 , R0) ;	162.822	A1 = ASHIFT A1 BY R0.L ;	167.038
R0.L = ONES R0 ;	166.103	R0.L = ASHIFT R0.H BY R0.L ;	167.085
A1 = ROT A1 BY R0.L ;	166.121	R0.H = ASHIFT R0.H BY R0.L (S) ;	167.137
R0.L = CC = BXOR (A0 , A1 , CC) ;	166.214	R0 = DEPOSIT (R0 , R0) (X) ;	167.158
R0.L = SIGNBITS A1 ;	166.218	R0 = EXTRACT (R0 , R0.L) (X) ;	167.178
R0 = PACK (R0.H , R0.H) ;	166.227	R0 = LSHIFT R0 BY R0.L ;	167.185
R0.L = CC = BXOR (A0 , R0) ;	166.342	R0 = EXTRACT (R0 , R0.L) (Z) ;	167.316
R0.L = SIGNBITS R0.H ;	166.366	R0.H = ASHIFT R0.H BY R0.L ;	167.319
R0.L = EXPADJ (R0.H , R0.L) ;	166.38	R0.L = LSHIFT R0.H BY R0.L ;	167.325
R0.L = SIGNBITS A0 ;	166.423	R0.H = ASHIFT R0.L BY R0.L (S) ;	167.331
R0 = PACK (R0.L , R0.H) ;	166.52	R0.L = CC = BXORSHIFT (A0 , R0) ;	167.337
R0.L = SIGNBITS R0 ;	166.548	R0 = DEPOSIT (R0 , R0) ;	167.35
R0 = PACK (R0.H , R0.L) ;	166.598	A0 = LSHIFT A0 BY R0.L ;	167.366
A0 = ROT A0 BY R0.L ;	166.641	R0 = ASHIFT R0 BY R0.L (V) ;	167.404
R0.L = EXPADJ (R0 , R0.L) (V) ;	166.643	R0.L = VIT_MAX (R0) (ASL) ;	167.433
A1 = LSHIFT A1 BY R0.L ;	166.647	A0 = ASHIFT A0 BY R0.L ;	167.476
R0.L = SIGNBITS R0.L ;	166.648	R0.L = VIT_MAX (R0) (ASR) ;	167.486
R0.L = LSHIFT R0.L BY R0.L ;	166.668	R0 = ASHIFT R0 BY R0.L (S , V) ;	167.52
R0.H = LSHIFT R0.L BY R0.L ;	166.753	R0 = LSHIFT R0 BY R0.L (V) ;	167.642
R0.L = EXPADJ (R0.L , R0.L) ;	166.77	R0 = ASHIFT R0 BY R0.L ;	167.805
R0.H = LSHIFT R0.H BY R0.L ;	166.793	R0 = ASHIFT R0 BY R0.L (S) ;	167.879
R0 = ROT R0 BY R0.L ;	166.835	R0 = VIT_MAX (R0 , R0) (ASR) ;	168.12
R0.L = ASHIFT R0.L BY R0.L ;	166.847	BITMUX (R0 , R1 , A0) (ASR) ;	168.453
R0 = PACK (R0.L , R0.L) ;	166.886	R0 = VIT_MAX (R0 , R0) (ASL) ;	168.477
R0.L = EXPADJ (R0 , R0.L) ;	166.899	BITMUX (R0 , R1 , A0) (ASL) ;	168.813
R0.L = ASHIFT R0.H BY R0.L (S) ;	166.98	A0 = BXORSHIFT (A0 , A1 , CC) ;	175.134

Table 17. Instruction base cost for dsp32shift group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
A1 = ROT A1 BY 0 ;	166.17	R0.H = R0.H >>> 15 ;	167.817
A0 = ROT A0 BY 0 ;	166.631	R0.L = R0.L << 0x0 (S) ;	167.831
A1 = A1 >> 0 ;	166.687	R0.L = R0.L >> 15 ;	167.84
A0 = A0 >> 31 ;	167.214	R0 = R0 >>> 31 ;	167.873
A1 = A1 >> 31 ;	167.267	R0.L = R0.H >>> 15 ;	167.878
A0 = A0 >> 0 ;	167.31	R0.L = R0.L >>> 15 ;	167.931
R0 = ROT R0 BY 0 ;	167.382	R0.L = R0.H << 0x0 (S) ;	167.956
A0 = A0 >>> 31 ;	167.401	R0.H = R0.L << 0x0 ;	167.989
A1 = A1 >>> 31 ;	167.55	R0.H = R0.L >>> 15 ;	168.144
R0.L = R0.H >> 15 ;	167.558	R0 = R0 >> 0 ;	168.161
R0.H = R0.L >> 15 ;	167.573	R0.H = R0.H << 0x0 (S) ;	168.225
R0 = R0 >> 15 (V) ;	167.575	R0.H = R0.L << 0x0 (S) ;	168.235
R0.H = R0.H << 0x0 ;	167.595	R0 = R0 << 0x0 (V) ;	168.369
R0.L = R0.L << 0x0 ;	167.662	R0 = R0 << 0x0 (S , V) ;	168.441
R0 = R0 >> 31 ;	167.747	R0 = R0 >>> 0 (S) ;	168.473
R0.L = R0.H << 0x0 ;	167.751	R0 = R0 >>> 0 (V) ;	168.528
R0.H = R0.H >> 15 ;	167.752		

Table 18. Instruction base cost for dsp32shiftimm group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0.L = W [I0] ;	167.3	W [I0 ++] = R0.H ;	182.105
R0.H = W [I0] ;	167.359	W [I0 ++] = R0.L ;	182.816
R0 = [I0] ;	168.933	R0 = [I0 ++] ;	182.821
W [I0] = R0.H ;	169.786	[I0 ++] = R0 ;	183.09
W [I0] = R0.L ;	169.788	[I0 --] = R0 ;	183.961
R0 = [I0 ++ M0] ;	169.835	R0.H = W [I0 --] ;	184.386
[I0] = R0 ;	170.452	R0.L = W [I0 --] ;	184.702
[I0 ++ M0] = R0 ;	171.82	W [I0 --] = R0.L ;	185.198
R0.L = W [I0 ++] ;	181.111	W [I0 --] = R0.H ;	185.227
R0.H = W [I0 ++] ;	181.162	R0 = [I0 --] ;	185.849

Table 19. Instruction base cost for dspLDST group

Instruction	Instruction Base Cost (mA)
R0 = 0 (X) ;	159.43
R0 = 0x0 (Z) ;	165.732
R0.H = 0 ;	165.804
R0.L = 0 ;	166.652

Table 20. Instruction base cost for LDIMMhalf group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 = B [P2] (X) ;	168.419	W [P2 ++] = R0 ;	178.226
R0 = W [P2] (Z) ;	168.573	R0 = [P2 ++] ;	178.954
R0 = B [P2] (Z) ;	168.589	P1 = [P2 ++] ;	179.075
R0 = W [P2] (X) ;	168.671	[P2 ++] = R0 ;	179.428
B [P2] = R0 ;	169.462	R0 = B [P2 --] (Z) ;	179.59
P0 = [P2] ;	169.707	B [P2 --] = R0 ;	180.202
R0 = [P2] ;	169.771	[P2 --] = R0 ;	180.371
W [P2] = R0 ;	170.594	R0 = W [P2 --] (Z) ;	181.297
[P2] = R0 ;	171.512	W [P2 --] = R0 ;	181.407
[P2] = P0 ;	174.106	R0 = [P2 --] ;	181.625
R0 = B [P2 ++] (Z) ;	175.602	R0 = W [P2 --] (X) ;	181.674
R0 = B [P2 ++] (X) ;	175.724	[P2 --] = P0 ;	181.782
B [P2 ++] = R0 ;	175.884	[P2 ++] = P0 ;	181.839
R0 = W [P2 ++] (Z) ;	177.058	R0 = B [P2 --] (X) ;	181.974
R0 = W [P2 ++] (X) ;	177.087	P1 = [P2 --] ;	182.254

Table 21. Instruction base cost for LDST group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 = B [P0 + 0] (Z) ;	168.041	R0 = B [P0 + -8] (X) ;	173.942
R0 = B [P0 + 0] (X) ;	168.357	R0 = W [P0 + -6] (Z) ;	173.963
R0 = W [P0 + 0] (Z) ;	168.447	R0 = W [P0 + -6] (X) ;	174.099
R0 = W [P0 + 0] (X) ;	168.476	[P0 + 0] = P0 ;	174.125
R0 = [P0 + 0] ;	169.38	B [P0 + -8] = R0 ;	175.726
B [P0 + 0] = R0 ;	170.312	W [P0 + -6] = R0 ;	176.509
W [P0 + 0] = R0 ;	170.879	[P0 + -4] = R0 ;	176.981
[P0 + 0] = R0 ;	171.744	[P0 + -4] = P0 ;	179.173
R0 = B [P0 + -8] (Z) ;	173.745		

Table 22. Instruction base cost for LDSTidxI group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 = W [P0 + 0x0] (X) ;	167.269	W [P0 + 0x0] = R0 ;	170.678
R0 = W [P0 + 0x0] (Z) ;	168.765	[P0 + 0x0] = R0 ;	171.655
R0 = [P0 + 0x0] ;	169.575	[P0 + 0x0] = P0 ;	172.658

Table 23. Instruction base cost for LDSTii group

Instruction	Instruction Base Cost (mA)
R0 = [FP + -128] ;	147.273
P0 = [FP + -128] ;	147.478
[FP + -128] = R0 ;	147.898
[FP + -128] = P0 ;	149.979

Table 24. Instruction base cost for LDSTiiFP group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 >>= 0x0 ;	161.501	R0 >>>= 0x0 ;	162.727
R0 <<= 0x0 ;	161.56	BITCLR (R0 , 0x0) ;	162.832
CC = ! BITTST (R0 , 0x0) ;	161.632	BITSET (R0 , 0x0) ;	163.235
CC = BITTST (R0 , 0x0) ;	161.685	BITTGL (R0 , 0x0) ;	163.763

Table 25. Instruction base cost for LOGI2op group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
R0 = [P0 ++ P0] ;	166.542	W [P1 ++ P0] = R0.L ;	168.632
R0.H = W [P0] ;	166.464	W [P1 ++ P0] = R0.H ;	169.832
R0.L = W [P0] ;	169.572	R0 = W [P2 ++ P0] (X) ;	166.763
[P0 ++ P0] = R0 ;	167.442	R0.H = W [P2 ++ P0] ;	167.652
W [P0] = R0.L ;	167.571	R0 = W [P2 ++ P0] (Z) ;	167.621
W [P0] = R0.H ;	169.821	R0.L = W [P2 ++ P0] ;	167.399

Table 26. Instruction base cost for LDSTpmod group

Instruction	Instruction Base Cost (mA)
LINK 0x0 ; UNLINK ;	148.591

Table 27. Instruction base cost for linkage group

Instruction	Instruction Base Cost (mA)
LSETUP (4,4) LC0=P5;	169.22

Table 28. Instruction base cost for LoopSetup group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
IDLE ;	129.527	RAISE 0x0 ;	155.934
EMUEXCPT ;	146.871	NOP ;	158.651
TESTSET (P0) ;	148.259	CSYNC ;	159.599
SSYNC ;	153.363	CLI R0 ;	160.543
STI R0 ;	155.118		

Table 29. Instruction base cost for ProgCtrl group

Instruction	Instruction Base Cost (mA)	Instruction	Instruction Base Cost (mA)
P0 -= P0 ;	159.146	P0 = P0 << 2 ;	160.834
P0 = P0 >> 2 ;	159.933	P0 = (P0 + P0) << 1 ;	160.881
P0 = (P0 + P0) << 2 ;	160.26	P0 = P0 >> 1 ;	161.296
P0 += P0 (BREV) ;	160.347		

Table 30. Instruction base cost for PTR2op group

Instruction	Instruction Base Cost (mA)
[-- SP] = R0 ; R0 = [SP ++] ;	146.962

Table 31. Instruction base cost for PushPop group

Instruction	Instruction Base Cost (mA)
[-- SP] = (R7:0 , P5:0) ; (R7:0 , P5:0) = [SP ++] ;	148.156
(R7:0) = [SP ++] ; [-- SP] = (R7:0) ;	148.361
[-- SP] = (P5:0) ; (P5:0) = [SP ++] ;	149.175

Table 32. Instruction base cost for PushPopMultiple group

Instruction	Instruction Base Cost (mA)
R0 = R0 ;	160.798

Table 33. Instruction base cost for RegREGMV group

Instruction	Instruction Base Cost (mA)
JUMP.S 0 /*0x0*/ ;	173.817

Table 34. Instruction base cost for UJUMP group

References

[1] *Developing Energy-Aware Strategies for the Blackfin Processor* (http://www.ece.neu.edu/groups/nucar/publications/hpec04.pdf). Steven VanderSanden, David R. Kaeli, Giuseppe Olivadoti, Richard Gentile, May 2004. Northeastern University.

[2] *Estimating Power for ADSP-BF533 Blackfin Processors (EE-229).* Rev 3, May 2007, Analog Devices, Inc.

[3] *VisualDSP++ 4.5 Users Guide*. Revision 2.0, April 2006, Analog Devices, Inc.

Document History

Revision	**Description**
Rev 1 – June 26, 2007 *by Brian Mack and Seth Molloy*	Initial revision.